

# Отчет по первой лабораторной работе по сегментации изображений

Дмитриев Леонид

30 марта 2023 г.

## Содержание

Постановка задачи	3
Описание данных	3
Описание метода решения	3
Преобразования изображения . . . . .	3
Построение признаковых описаний по контурам фишек . . . . .	4
Определение типа фишки по признаковому описанию . . . . .	5
Описание программной реализации	5
Эксперименты	5
Выводы	5

## Постановка задачи

Задача заключается в разработке и реализации программы для работы с изображениями фишек Тантрикс, обеспечивающей функционал, требуемый для класса сложности Intermediate:

1. Определение типа фишки по входному изображению, содержащему одну фишку.
2. Определение типа и расположения всех фишек, находящихся во входном изображении.

## Описание данных

Игровой набор Тантрикс состоит из 10 фишек, представленных на рисунке (файл Dozen\_0.bmp).



Рис. 1: Пример с правильной классификацией всех 10 типов имеющихся фишек.

Каждая фишка представляет собой правильный шестиугольник черного цвета, на котором изображены сегменты трёх линий синего, красного и жёлтого цветов.

В простых случаях на картинках изображена одна фишка, а в более сложных несколько.



Рис. 2: Примеры входных изображений: для задачи 1 (слева) и для задачи 2 (в центре и справа).

## Описание метода решения

В первую очередь было замечено, что решение задачи определения типа одной единственной фишки в кадре вкладывается в решение задачи определения типа и местоположения всех фишек на кадре. Поэтому далее решалась только более общая задача.

Далее описаны основные моменты алгоритма, решающего поставленную задачу.

## Преобразования изображения для поиска контуров

Для определения контуров необходимо подсветить темные объекты на изображении, а фон обнулить. Сначала картинка преобразовывается в черно-белый спектр. Далее используется гауссов фильтр для сглаживания

изображения. А затем применяется инвертированная бинаризация (так чтобы темная поверхность фишек стала максимально яркой, а все остальное черным).

Для определения типа фишки необходима информация о цветных контурах внутри него. Сначала картинка преобразовывается в тип HSV, а затем по определенному диапазону тона, насыщенности, значений находится маска нужных цветов (диапазон определен экспериментально).

В итоге к построенным маскам применяется функция поиска контуров из библиотеки `opencv`, с помощью которой для каждого шестиугольника и цветного элемента внутри фишек находится обход точек по границе.



Рис. 3: Примеры определения контуров на преобразованном изображении.

После поиска контуров могли возникнуть шумовые контуры как фишек, так и цветных компонент. Они удаляются по разным правилам, например: по площади относительно остальных фигур или по расположению внутри других фигур.

## Построение признаков описаний по контурам фишек

Полученные контуры необходимо преобразовать в удобный для анализа вид.

В случае правильного шестиугольника полезно найти 6 угловых точек. Для этого сначала удаляются точки образующие с соседними угол, близкий к 180 градусам. После этого остаются только точки находящиеся близко к углам. Для них применяется алгоритм кластеризации с 6 кластерами, центроиды которого и являются приближенными угловыми точками.

Контуры цветных компонент для упрощения покрываются минимальными прямоугольниками.

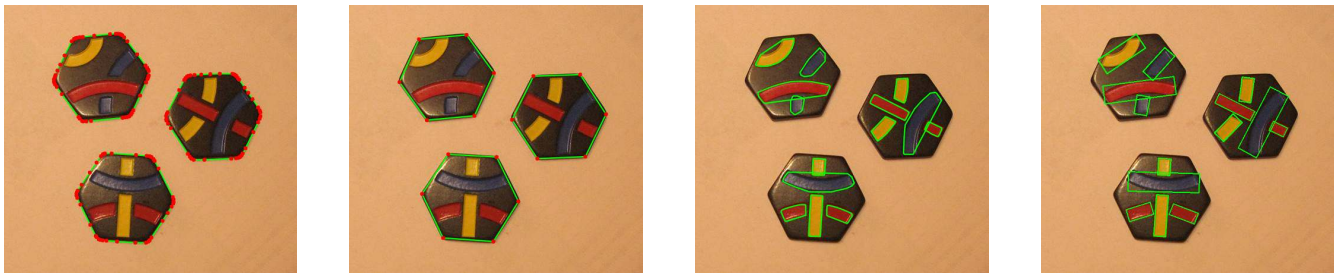


Рис. 4: Примеры определения контуров на преобразованном изображении.

Можно заметить, что каждая цветная компонента соединена ровно с двумя ребрами шестиугольника (причем эти ребра принадлежат только этой компоненте). Также каждая компонента либо один раз пересечена другой компонентой, либо цела.

Зная для каждого цвета соответствующие ему стороны шестиугольника и факт пересеченности можно однозначно определить тип фишки.

Далее можно заметить, что центры меньших сторон прямоугольников соотв. цветным компонентам ближе всего к нужному центру сторон шестиугольника. А факт пересеченности определяется по кол-ву связанных компонент данного цвета.

По данным признакам строится исчерпывающее описание наблюдаемой фишки.

Но в связи с различными ошибками на любом из этапов данное описание может оказаться неполным или некорректным. Одним из способов привнесения большей стабильности служит алгоритм восстановления одного из цветов по двум другим (при условии, что они определены корректно).

Очевидно, что при определении сторон шестиугольника для двух цветов и факта пересеченности этих двух цветов, третьему цвету останется только одно возможное состояние.

## Определение типа фишки по признаковому описанию

Имея описание фишки, полученное на предыдущем шаге, можно легко установить тип фишки, если заранее описать все имеющиеся типы в таком же формате и просто сравнить с каждым из них.

## Описание программной реализации

Для запуска программы решения:

1. `python main.py --mode single --files <filepath>`

Программа прочтет картинку из `<filepath>` и напечатает номер типа фишки и его представление в указанном выше формате.

2. `python main.py --mode group --files <src_filepath> <dest_filepath> [--show False]`

Программа прочтет картинку из `<src_filepath>` и запишет картинку такого же размера в `<dest_filepath>`. В результирующей картинке поверх каждой определенной фишки будет написан ее тип. Так же опционально возможно указать, выводить ли окно с результатом.

## Эксперименты

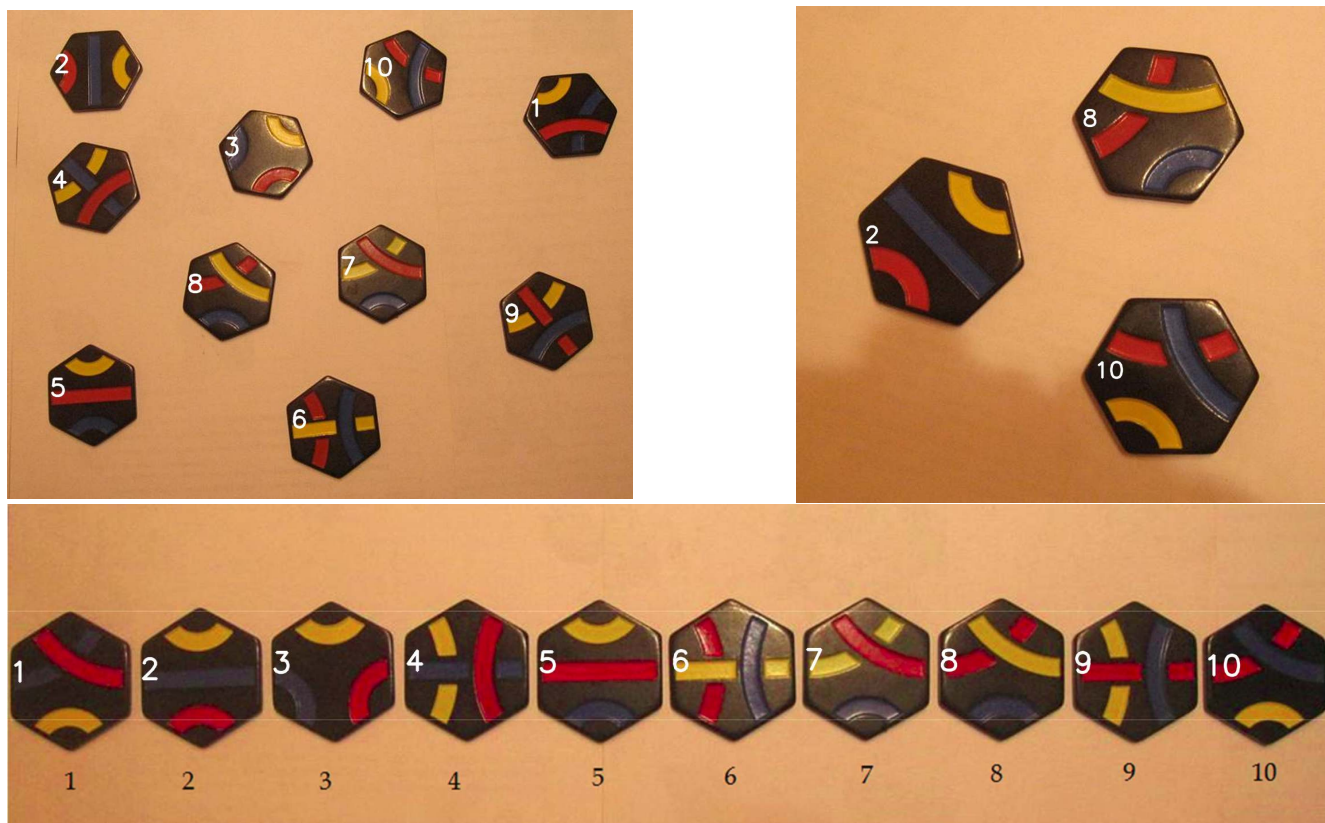


Рис. 5: Примеры результатов программы.

## Выводы

Была решена задача распознавания фишек формы правильного шестиугольника с расположенными на нем линиями разных цветов и реализована соответствующая программа.